Michigan Math. J. 58 (2009)

Implementing Gradient Descent Decoding

ROBERT A. LIEBLER

In memory of D. G. Higman

1. Introduction

Many communication channels accept as input binary strings and return output strings of the same length that have been altered in an unpredictable way. To compensate for these "errors", redundant data is added to messages before they enter the channel. The task of a decoding algorithm is to *reconstruct sent message(s)* (i.e., to decode) the channel output.

There are several critical attributes of a decoding algorithm. The *design complexity* is a measure of the effort required to design and implement an instance of the algorithm. The *implementation cost* is a measure of the time and space resources required to decode received sequences once the algorithm is implemented. The *apparent accuracy* of an algorithm is a measure of its ability to actually identify all most likely *sent messages* (codewords) for a given received sequence in practice. The *proven accuracy* is a measure of the algorithm's certified ability to correctly move from received sequence to nearest codeword without failure of any kind. A decoding algorithm is *optimal* if it correctly identifies the most likely channel error for each possible received sequence for which there is a unique such error.

Error correcting codes have uses beyond communication channels, and different applications have different decoding accuracy requirements. An important challenge is to find methods of proven accuracy that perform optimal decoding and have implementation cost low enough to be practical with codes that include very long words.

As the foundations of coding theory developed in the 1950s, there arose a method called *syndrome decoding* that is provably optimal but possibly costly to design and implement. Starting with "turbo" codes and then with *low-density parity check* (LDPC) codes, iterative decoding has attracted wide interest in the past 15 years. In each case, these terms refer to both encoding and decoding algorithms. The chief advantage of these new methods is their simplicity of decoding implementation. By Shannon's fundamental work, communication at rates approaching channel capacity requires long codes, which are practical only with low-complexity decoding. Just such communication has been achieved in practice by turbo codes and proven by extensive simulation under strong decoder assumptions for LDPC

Received September 11, 2007. Revision received May 2, 2008.

codes [9]. Unfortunately, a theoretical understanding of the reasons for the success of these algorithms has been elusive. In fact, it has been said that iterative decoding has "moved coding theory toward an experimental science" [3].

However, Richardson and Urbanke [10] have recently given an excellent readable analysis of "belief propagation" decoding performance for the binary erasure channel (the simplest nontrivial channel). There has also developed a theoretical understanding of why LDPC decoding sometimes fails to converge. A LDPC decoder comes equipped with a specific parity check matrix, the construction of which seems to be an art form involving substantial choice beyond the underlying code. There are many examples of the same code having different parity check matrices that yield very different iterative decoding success. The *Tanner graph* associated to a given parity check matrix is bipartite, with vertices labeled by rows (positions in the received sequence) and columns (syndrome positions). Edges of the Tanner graph are determined by the 1s in the parity check matrix.

It is now understood that when the Tanner graph has a graph covering that corresponds to a different code, the *message passing* decoding algorithm "becomes confused" because it cannot distinguish authentic codewords from *pseudo-codewords* that arise from the (graph-theoretic) cover [6]. Put another way, pseudo-codewords interfere with convergence because the decoding algorithm is "local" and doesn't distinguish between the Tanner graph and a covering graph. Any exclusively local algorithm faces this difficulty.

This study began with two questions: "Exactly what information, if any, does LDPC decoding use that classical syndrome decoding does not?" and "What is a reasonable global measure of decoding progress that could be used to avoid pseudo-codewords?" Very simple answers have emerged. Iterative decoding makes systematic use of "characteristic crossing" maps whereas classical texts have only one such map. It also leads to an answer for the second question—but in a new way, different from its role in classical syndrome decoding algorithms. *Hamming weight* counts the number of 1s in a binary sequence; thus it is a function whose domain has characteristic 2 but whose range is the natural numbers.

In this paper it is shown how to construct a gradient function for any linear code on a *binary symmetric channel* (BSC). (However, the basic ideas are more widely applicable.) There results an optimal iterative decoding algorithm with proven accuracy. Not surprisingly, this explicit construction has no implementation complexity advantage over classical syndrome decoding for an arbitrary code.

There are, however, many specific examples where the construction can be greatly simplified without sacrificing proven accuracy. In particular, it is possible to reduce the implementation cost of some highly geometric codes. In all cases the algorithm design includes construction of a function that increases with respect to (but does not necessarily coincide with) *most likely error Hamming weight* (coset-leader Hamming weight).

All of this leads a bigger question, for which there is currently no clear answer. What attributes of a code contribute to it having an easily recognized and simply implemented gradient function for coset-leader Hamming weight? My personal

inclination is to follow Professor Higman's interest in highly symmetric codes associated with rich combinatorial structures and hope that they are more easily analyzed.

The merits of gradient descent decoding have been recognized by others. For example, Lucas, Bossert, and Breitbach [8] give a similar algorithm and even a similarly constructed function, but the actual functions used are not monotonic with respect to Hamming weight and decoding convergence is uncertain. More recently, Justesen, Hørholdt, and Hjaltason [4] give a closely related algorithm based on a function they call "syndrome weight"; they show that it does have the critical monotonicity property for the [73, 45] code arising from the projective plane of order 8, but they have no general construction. Kelley and Sridhara [5] explore how the nonuniqueness of the parity check matrix might be exploited to improve performance of bit-flip decoding.

The "gradient-like decoding" algorithm of Ashikhmin and Barg [2] is similar to ours in some ways but is actually quite different. It stays entirely in one coset of the code and systematically seeks out a coset leader. For us, the sequence {kth coset leader's Hamming weight} forms a strictly decreasing sequence of nonnegative integers. Thus, the complexity of our algorithm is a linear function of most likely error weight.

This paper ends with an example for which "belief propagation" is ineffective [6]. Details are given for constructing a new (weighted) parity check matrix whose syndrome weight equals twice the coset-leader Hamming weight. This example is somewhat independent of the earlier material and could be a starting point for some readers.

Judy Walker's helpful questions and patient encouragement is gratefully acknowledged.

2. Gradient Function Construction

A binary $[n, k]$-code $C \leq \mathbb{F}_2^n$ is an additive subgroup of order 2^k in the row space $\mathbb{F}_2^n$. The matrix $H \in \mathrm{Mat}_{s,n}(\mathbb{F}_2)$ is a *parity check matrix* for C provided $\mathbf{m}H^T = \mathbf{0}$ if and only if $\mathbf{m} \in C$. (In particular, $s > n - k$ is allowed.) Parity check matrices represent a triumph of classical decoding because they effectively separate the encoded message from the channel error. Indeed if $\mathbf{m} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in C$ then the *syndrome* $\mathbf{m}H^T = (\mathbf{c} + \mathbf{e})H^T = \mathbf{e}H^T$ is independent of $\mathbf{c} \in C$ and depends only on $\mathbf{e}$. The C-coset $\overline{\mathbf{m}} := \{\mathbf{m} : \mathbf{m}'H^T = \mathbf{m}H^T\}$ determined by the syndrome $\mathbf{m}H^T$ is an element of the additive group $\mathbb{F}_2^n/C$.

Within each C-coset, the sequences having minimal weight are called *coset leaders.* Define $\mathrm{wt}(\overline{\mathbf{m}})$ to be the Hamming weight of (any) coset leader of $\overline{\mathbf{m}}$. The task of an *optimal decoder* is to decide, based on the syndrome $\mathbf{m}H^T$ only, whether the associated coset $\overline{\mathbf{m}}$ has a unique coset leader $\mathbf{e}$ and, if so, to return $\mathbf{c} = \mathbf{m} - \mathbf{e}$ $(= \mathbf{m} + \mathbf{e}) \in C$. The classical syndrome decoder is essentially a table that matches syndromes to the associated coset leaders. For fixed rate k/n, the size of such a table grows exponentially as a function of codeword length n.

One feature that separates "gradient descent" from other decoding methods is that this task is broken into smaller steps. The gradient function allows the decoder to first find a Hamming neighbor $\mathbf{m}'$ of $\mathbf{m}$ (Hamming distance $d(\mathbf{m}', \mathbf{m}) = 1$) such that $\mathrm{wt}(\overline{\mathbf{m}'}) < \mathrm{wt}(\overline{\mathbf{m}})$. Then one replaces $\mathbf{m}$ with $\mathbf{m}'$ and iterates until $\mathrm{wt}(\overline{\mathbf{m}}) = 0$. Thus there is no need to identify the actual error $\mathbf{e}$.

Of course, this method requires a *gradient function* $\gamma : \mathbb{F}_2^n/C \to \mathbb{Z}$ such that $\gamma(\mathbf{m})$ is a strictly increasing function of $\mathrm{wt}(\overline{\mathbf{m}})$, and such functions have not been widely available. In fact, this paper's main contribution is to present the first such construction method. We begin by ignoring efficiency and only later investigate what is really necessary for this kind decoding algorithm to be efficient.

The argument uses the characteristic-0 properties of a certain incidence matrix. Let $\mathcal{P}$ be the set of points of the projective space $\mathrm{PG}(\mathbb{F}_2^n/C)$ (which can be identified with the set of nonzero elements of $\mathbb{F}_2^n/C$ since the field here is $\mathbb{F}_2$), and let $\mathcal{H}$ be the set of hyperplanes of $\mathrm{PG}(\mathbb{F}_2^n/C)$. Also let N be the incidence matrix of the relation $\mathcal{N} \subset \mathcal{P} \times \mathcal{H}$ for which $(p, h) \in \mathcal{N}$ exactly when p is *not* contained in the hyperplane h. Note that the rows of N are indexed by $\mathcal{P}$ and the columns by $\mathcal{H}$. Thus the (p, q)-entry of NN^T is the number of hyperplanes containing neither p nor q:

$$N^T N = \lambda(I + J) = NN^T, \tag{1}$$

where $\lambda = 2^{n-k-3}$, I is the identity matrix, and J is the all-1 matrix of size(s) $2^{n-k} - 1$.

Cautiously introduce (these need not associate with other maps) "characteristic crossing" functions $\blacktriangle : \mathbb{F}_2^s \to \mathbb{Z}^s$ and $\blacktriangledown : \mathbb{Z}^s \to \mathbb{F}_2^s$, where s is determined by context and the spaces may also be matrix spaces. The map $\blacktriangledown$ is reduction modulo 2, but the map $\blacktriangle$ replaces binary 0 and 1 by the same symbols regarded as integers; both maps act coordinate-wise. These maps will be used with matrices and vectors, themselves regarded as maps, acting on the right. For example, the *binary* matrix $(\blacktriangle N^T)\blacktriangledown$ with rows indexed by all $2^{n-k} - 1$ points and columns is indexed by all $2^{n-k} - 1$ hyperplanes of $\mathrm{PG}(\mathbb{F}_2^n/C)$. For each hyperplane h there is unique nonzero $h^\perp \in C^\perp = \{x \in \mathbb{F}_2^n \mid cx^T = 0 \text{ for all } c \in C\}$ such that

$$\mathbf{m}{h^\perp}^T = N^T_{(\overline{\mathbf{m}}, h)} \pmod 2 \quad \text{for all } \mathbf{m} \in \mathbb{F}_2^n.$$

In other words, the hth column of $N^T\blacktriangledown$ is the *table of values* of the $\mathbb{F}_2$-linear functional mapping $\mathbb{F}_2^n/C$ to $\mathbb{F}_2$ determined by $h^\perp \in C^\perp \setminus \{0\}$.

Let $H^T \in \mathrm{Mat}_{2^{n-k}-1,n}(\mathbb{F}_2)$ have the same column labels as N^T and have as entries in its hth column the coordinates of $h^\perp \in C^\perp \subset \mathbb{F}_2^n$. Observe that H^T is the widest possible parity check matrix for C that does not have repeated columns.

Suppose $\mathbf{m} \in \mathbb{F}_2^n$ has coset leader $\mathbf{e} \neq \mathbf{0}$, so $\overline{\mathbf{m}} = \bar{\mathbf{e}} \in \mathcal{P}$. Then $\mathbf{m}H^T = \mathbf{e}H^T$ and so $(\mathbf{m}H^T)\blacktriangle = (\mathbf{e}H^T)\blacktriangle$ has hth coordinate $1 \in \mathbb{Z}$ if and only if the hyperplane h does not contain $\bar{\mathbf{e}}$. Therefore $(\mathbf{m}H^T)\blacktriangle$ is the $\bar{\mathbf{e}}$th row of N^T. Thus

$$(\mathbf{m}H^T)\blacktriangle = \chi(\bar{\mathbf{e}})N^T,$$

where $\chi(\bar{\mathbf{e}})$ denotes the (integer-valued) characteristic function of $\bar{\mathbf{e}}$: the row with 1 in the $\bar{\mathbf{e}}$th position and 0 elsewhere.

THEOREM 1. *Let* $\mathbf{wt}$ *be vector with coordinates labeled by* $\mathcal{P}$ *having* $\overline{\mathbf{m}}$th *coordinate equal to the Hamming weight* $\mathrm{wt}(\overline{\mathbf{m}})$ *of a coset leader of the* C*-coset* $\overline{\mathbf{m}} = \mathbf{m} + C$; *that is,*

$$\mathbf{wt} := \sum_{\overline{\mathbf{m}} \in \mathcal{P}} \mathrm{wt}(\overline{\mathbf{m}})\overline{\mathbf{m}} \in \mathbb{Z}\mathcal{P}.$$

Then the function γ *defined by* $\gamma(\mathbf{m}) := ((\mathbf{m}H^T)\blacktriangle)N\,\mathbf{wt}^T$ *is a gradient function for coset Hamming weight.*

Proof. By equations (2) and (1),

$$\begin{aligned}\gamma(\mathbf{m}) := ((\mathbf{m}H^T)\blacktriangle)N\,\mathbf{wt}^T &= (\chi(\mathbf{e})N^T)(N\,\mathbf{wt}^T) = \chi(\mathbf{e})(N^TN)\,\mathbf{wt}^T \\ &= \chi(\mathbf{e})2^{n-k-3}(I_B + J_B)\,\mathbf{wt}^T = 2^{n-k-3}(\mathrm{wt}(\mathbf{e}) + \mathbf{j}\,\mathbf{wt}^T) \\ &= 2^{n-k-3}\,\mathrm{wt}(\mathbf{e}) + K,\end{aligned}$$

where $K = 2^{n-k-3}\sum_{\bar{\mathbf{e}}\in\mathcal{P}} \mathrm{wt}(\bar{\mathbf{e}})$. Thus $\gamma(\mathbf{m})$ is an increasing function of $\mathrm{wt}(\bar{\mathbf{e}})$. □

It may be of theoretical interest to point out that, where the argument just given uses Hamming weight of $\bar{\mathbf{e}}$, one could instead use any numerical tag. In particular, one could tag $\bar{\mathbf{e}}$ with the number whose binary expansion is the concatenation of *all* coset leaders of $\bar{\mathbf{e}}$. In this case, the theorem gives a complete list decoder without any iteration.

One of many algorithms used with LDPC codes is called "bit flipping" [4; 7], where a sparse 0,1 matrix H^T is used as a parity check matrix (among other things). The *syndrome weight* of a received sequence $\mathbf{m}$ is defined to be $\delta(\mathbf{m}) := ((\mathbf{m}H^T)\blacktriangle H)\mathbf{j}^T$. This algorithm searches the received sequence's Hamming neighbors to find $\mathrm{argmax}_{\mathbf{m}'}(\delta(\mathbf{m}) - \delta(\mathbf{m}')$; it then replaces $\mathbf{m}$ with $\mathbf{m}'$ and iterates.

Notice that the syndrome weight function δ differs from our γ only in the last two terms ($N\,\mathbf{wt} \neq H\mathbf{j}^T$). These functions almost coincide when $H\blacktriangle = N$ equals the incidence matrix of points and hyperplanes of a projective geometry over $\mathbb{F}_2$. In this case the code is a perfect Hamming code, and our method was once known as "majority logic decoding".

It is natural to ask how many of the columns of the H^T in the preceding construction can be omitted without destroying the gradient nature of γ. All that is necessary is that $\gamma(\mathbf{x}) > \gamma(\mathbf{y})$ whenever the coset-leader Hamming weight of $\mathbf{x}$ is greater than that of $\mathbf{y}$ and $x - y$ has Hamming weight 1. Just such functions γ arise for a variety of codes based on finite geometries, including the code of the projective plane of order 4 studied by Smarandache and Vontobel [11]. They have also been constructed for the $[85, 68, 6]$ and $[85, 60, 8]$ Lander codes (associated with incidence matrix elementary divisors) based on points of $\mathrm{PG}(3, 4)$. It remains to be seen how general these enticing results really are.

In order to underscore this question's importance, we estimate the complexity of implementing gradient descent decoding for a gradient function γ as described in Theorem 1 except that H has only m columns (and only N corresponding rows). We also allow an arbitrary fixed integer $\mathbf{wt}$ vector.

Let s and t be the (maximum) row and column sums of $H\blacktriangle$, the new parity check matrix made integral. Let α be the cost of multiplying two binary numbers (when the result is not known in advance because one of them is 0) and then adding the result to a third binary number. Also let β be the cost of adding/subtracting two integers. The expected cost of decoding a received sequence also depends on the (bitwise) error rate p of the channel.

The expected decoding cost has two types of terms. The cost of computing $\gamma(\mathbf{m})$ is always present at $\alpha nst + \beta m$. If there are no channel errors, then this is the only cost. The expected distribution of channel errors is uniform—every $\lceil n/np \rceil$ positions—so the expected cost of locating the first channel error involves the cost of computing the *change* in $\gamma(\mathbf{m})$ when $\mathbf{m}$ is replaced by each of its first $\lceil 1/p \rceil$ Hamming neighbors. Each such computation involves adding (if the change is to 1) or subtracting (if the change is to 0) the sum of s entries from the *weight vector* N **wt** to $\gamma(\mathbf{m})$. Thus the average cost of finding the first error is $\beta s/p$, and this cost occurs with probability p. The average cost of finding the second error is the same as the first because the sequence of positions visited while finding the first do not need to be revisited. The total decoding cost is thus bounded above by $s(\alpha nt + \beta(m + \lceil 1/(1-p) \rceil)$.

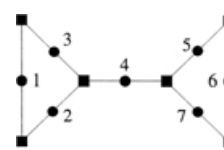
Since n, p, α, β are given and since β is surely greater than α, a sparse matrix H with as few columns as possible is what is needed. In particular, linear time complexity would require $s < K_s n$ and $m < K_m n$ for some fixed constants K_s and K_m.

3. An Example

Koetter, Li, Vontabel, and Walker [6] provide a simple but useful example to illustrate how pseudo-codewords form an obstruction to standard LDPC methods.

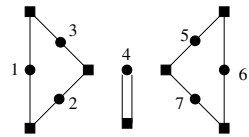
Consider the [7, 2] code associated with the indicated Tanner graph: received sequence symbols are numbered, and check symbols are marked with filled squares. This graph has the dihedral group of order 8 as symmetry group (generated by the sequence bit permutations (17)(2536) and (23)). Orbit representatives, orbit size, and syndrome weight of the 31 coset leaders are as follows.

	1001000 2 4	
1000000 2 2	1000100 4 4	1001100 4 4
0100000 4 2	1000010 1 4	1001010 1 6
0001000 1 2	0101000 4 2	0101100 4 2
	0100100 4 4	



Observe that the last sequences in the second and third column are Hamming neighbors yet the higher-weight one has lower syndrome weight. Therefore, syndrome weight is not an increasing function of Hamming weight for this LDPC code. Indeed, a bit-flipping algorithm based on this syndrome weight can fail to converge, with indefinite oscillation about the pseudo-codeword (1112111).

When the construction in Theorem 1 is applied to this code, the 31 possible parity checks are weighted $24^1\ 20^6\ 16^{24}$. The gradient property is unaffected by dividing these weights by 4 and subtracting 4. The indicated Tanner multi-graph emerges.



Syndrome weight for this parity check matrix equals 2 times the coset-leader Hamming weight. The bit-flipping algorithm based on this syndrome always converges in at most three steps. Note also that this generalized Tanner graph supports the full symmetry group $S_3 \wr S_2$ of the code.

References

[1] E. Agrell, *Voronoi regions for binary linear block codes,* IEEE Trans. Inform. Theory 42 (1996), 310–316.

[2] A. Ashikhmin and A. Barg, *Minimal vectors in linear codes,* IEEE Trans. Inform. Theory 44 (1998), 2010–2017.

[3] A. R. Calderbank, *The art of signaling: Fifty years of coding theory,* IEEE Trans. Inform. Theory 44 (1998), 2161–2195.

[4] J. Justesen, T. Høholdt, and J. Hjaltason, *Iterative list decoding,* Proceedings of IEEE information theory workshop on coding and complexity (M. J. Dinneen, U. Speidel, D. Taylor, eds.), pp. 90–93, IEEE, New York, 2005.

[5] C. Kelley and D. Sridhara, *Pseudocodewords of Tanner graphs,* IEEE Trans. Inform. Theory 53 (2007), 4013–4038.

[6] R. Koetter, W. W. Li, P. O. Vontobel, and J. L. Walker, *Pseudo-codewords of cycle codes via Zeta functions,* Proceedings of IEEE information theory workshop (San Antonio), 2004.

[7] Y. Kou, S. Lin, and M. P. C. Fossorier, *Low density parity check codes based on finite geometries: A rediscovery and new results*, IEEE Trans. Inform. Theory 47 (2001), 2711–2736.

[8] R. Lucas, M. Bosset, and M. Breitbach, *On iterative soft-decision decoding of linear binary block codes and product codes,* IEE J. Sel. Areas Commun. 16 (1998), 276–296.

[9] D. J. C. MacKay and R. M. Neil, *Near Shannon limit performance of low density parity check codes,* Elec. Lett. 32 (1996), 1645–1646.

[10] T. Richardson and R. Urbanke, *Modern coding theory,* Cambridge Univ. Press, Cambridge, 2008.

[11] R. Smarandache and P. O. Vontobel, *Pseudo-codeword analysis of Tanner graphs from projective and Euclidean planes,* IEEE Trans. Inform. Theory 53 (2007), 2376–2393.

Department of Mathematics
Colorado State University
Fort Collins, CO 80523

liebler@math.colostate.edu